Applied Genome Research

RNA-Seq: RNA to reads to counts

205048 & 205049

Boas Pucker

RNA-Seq

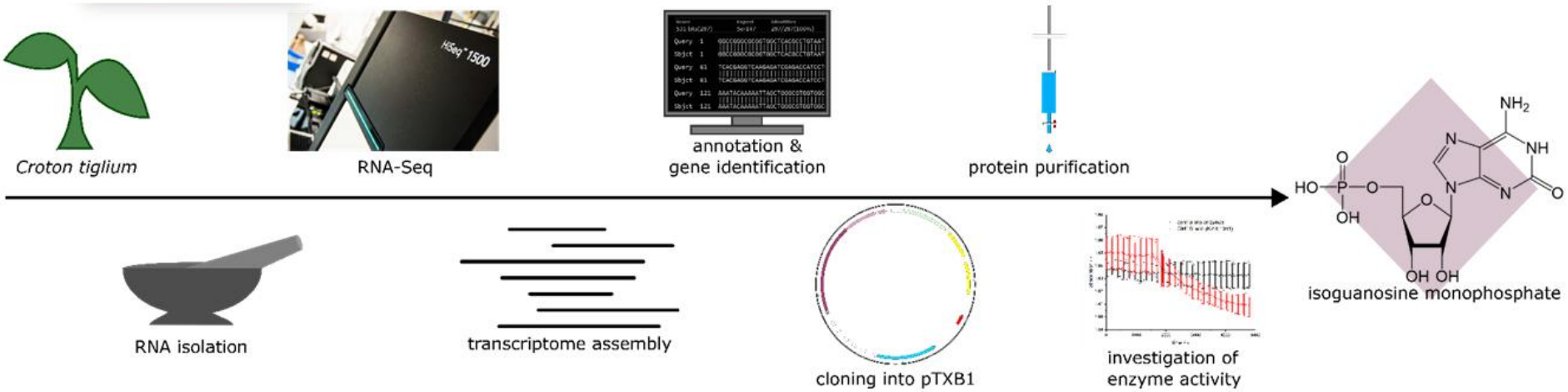
- Analysis of transcriptome via sequencing of cDNA (NOT RNA!)

RNA isolation ➡ cDNA amplification ➡ Library preparation ➡ Sequencing ➡ Data analysis

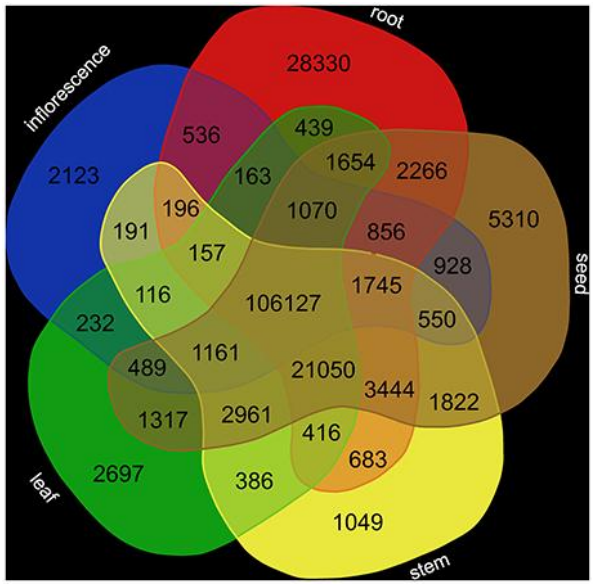
RNA-Seq applications

- Gene expression analysis
  - Many ‘tags’ are required => single end sequencing
  - PE sequencing would be beneficial for specific read mapping
  - Comparison of genotypes / conditions
- *De novo* transcriptome assembly
  - Samples from different tissues/conditions are used to increase diversity
  - PE sequencing is used to improve assembly continuity
  - Identification of novel transcripts (genes)
  - Analysis without genomic reference sequence

Application Example



(Next topic)



(iGEM Bielefeld-CeBiTec 2017, https://doi.org/10.3389/fmolb.2018.00062)

Tissue-specific quantification of transcript abundance

Microarray vs. RNA-Seq

	Microarray	RNA-Seq
Expression quantification	+	+
Detection of new transcripts	-	+
Dynamic range	-	+
Costs	?	?

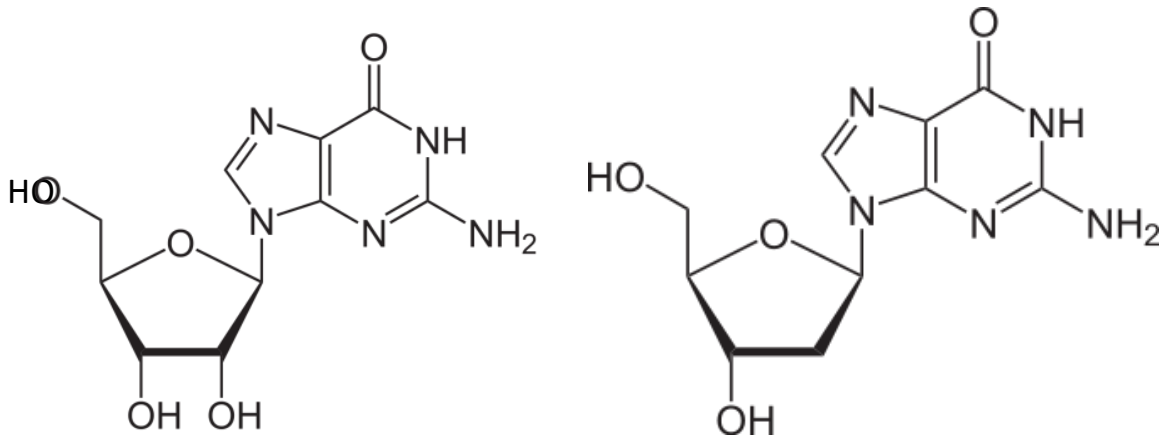
I) Gene expression analysis

1. Samples from different conditions / genotypes
2. Read generation (sequencing)
3. Read mapping
4. Read counting
5. Statistical analysis of differences

II) *De novo* transcriptome assembly

- Calculation of statistics and quality assessment
  - Contig length distribution
  - Unigene distribution
  - Read coverage depth
- Functional annotation
  - GO categories
  - Sequence similarity
- Differentially expressed genes (transcript abundance)
- GO or pathway enrichment analyses

RNA vs. DNA



(wikipedia.org)

RNA	DNA
Ribose sugar	Deoxyribose sugar
One strand	Two strands (double-helix)
C, G, A, U	C, G, A, T

Different types of RNA

- rRNA (ribosomal RNA) = essential component of ribosomes
- tRNA (transfer RNA) = delivers amino acids to ribosome and allows translation of mRNAs
- **mRNA (messenger RNA) = encodes peptide sequences**
- miRNA (micro RNA) = involved in regulation of gene expression
- ncRNA (non-coding RNA)
- …

RNA isolation

- Optimal protocol depends on downstream application, tissue, and species
- Classical approach: Trizol-based
- Our favorite kit for *A. thaliana*: NucleoSpin® RNA Plant
- Our favorite kit for *V. vinifera*: Spectrum™ Plant Total RNA

DNA contamination

- Separation between nucleic acids is not perfect
- Removal of DNA is important (DNase I treatment)
- gDNA and cDNA cannot be distinguished in later steps
- gDNA contaminations can prevent an *de novo* transcriptome assembly and influence expression analysis studies
- Deep sequencing will reveal even very small amounts of DNA

EXERCISE

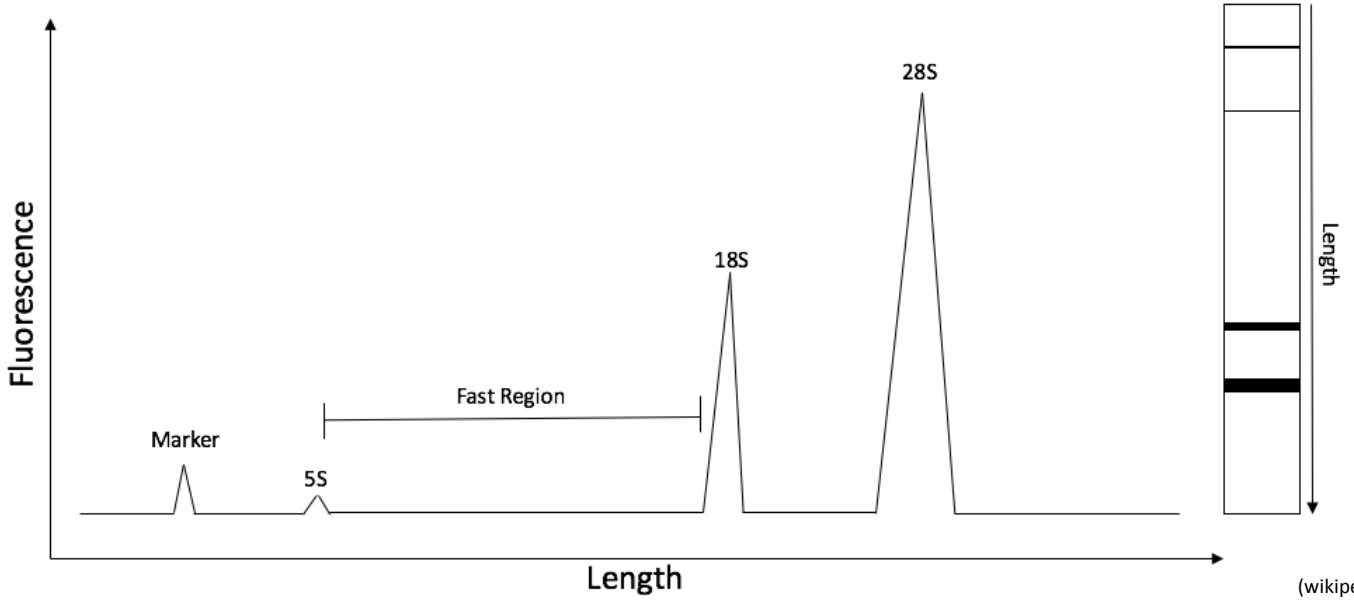
- Search the internet for protocols to isolate RNA from animals, plants or bacteria for RNA-Seq!

RNA quality control

- RNA agarose gel
- Nanodrop
- RIN (RNA integrity number)

RIN (RNA integrity number)

- Number between 1 (worst) and 10 (best)
- Ratio between area under 18S +28S rRNA peaks and the total area under the graph
- Height of 28S rRNA peak (more instable)



(wikipedia.org)

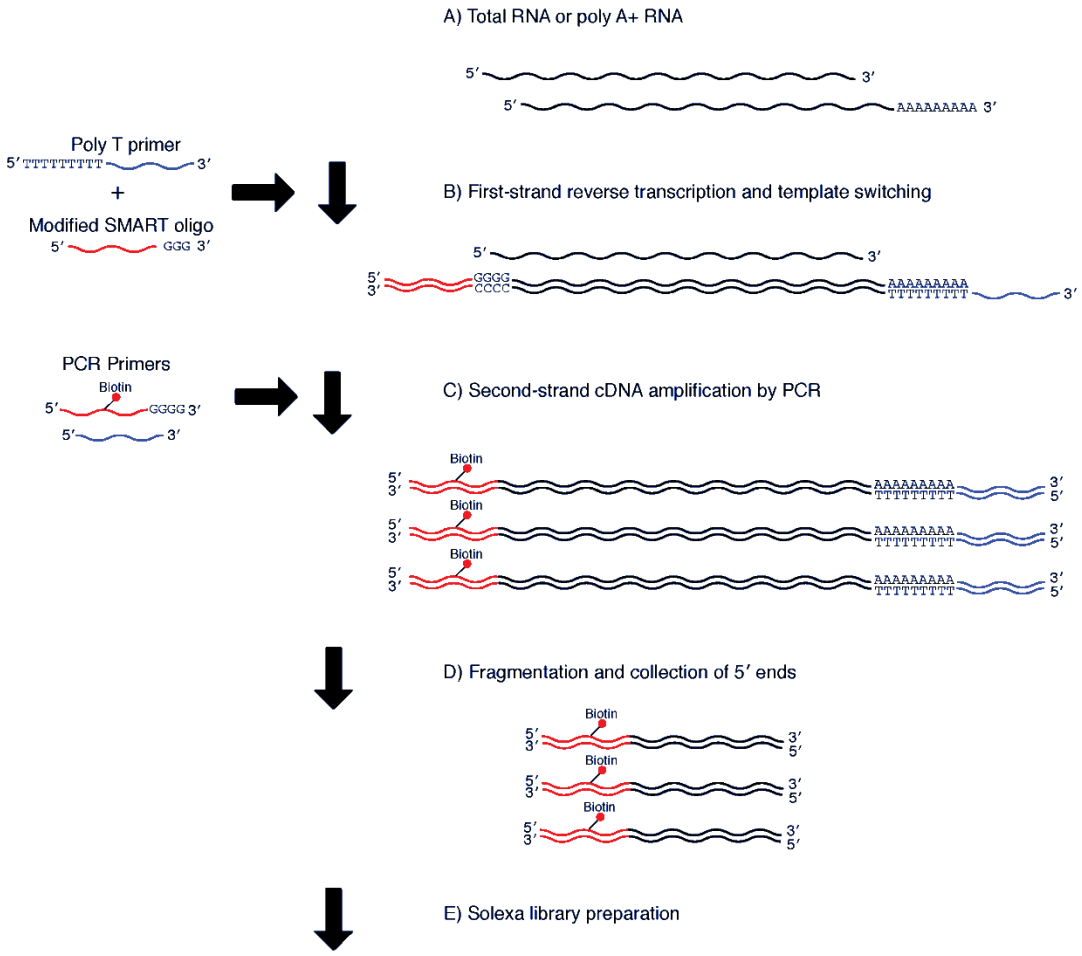
RNA-Seq library construction

- rRNA depletion (prokaryotes) or mRNA enrichment (eukaryotes)
- Fragmentation of RNA to achieve desired fragment length
- Reverse transcription into cDNA
- Adapter ligation and processing like genomic DNA libraries

Concept of rRNA depletion

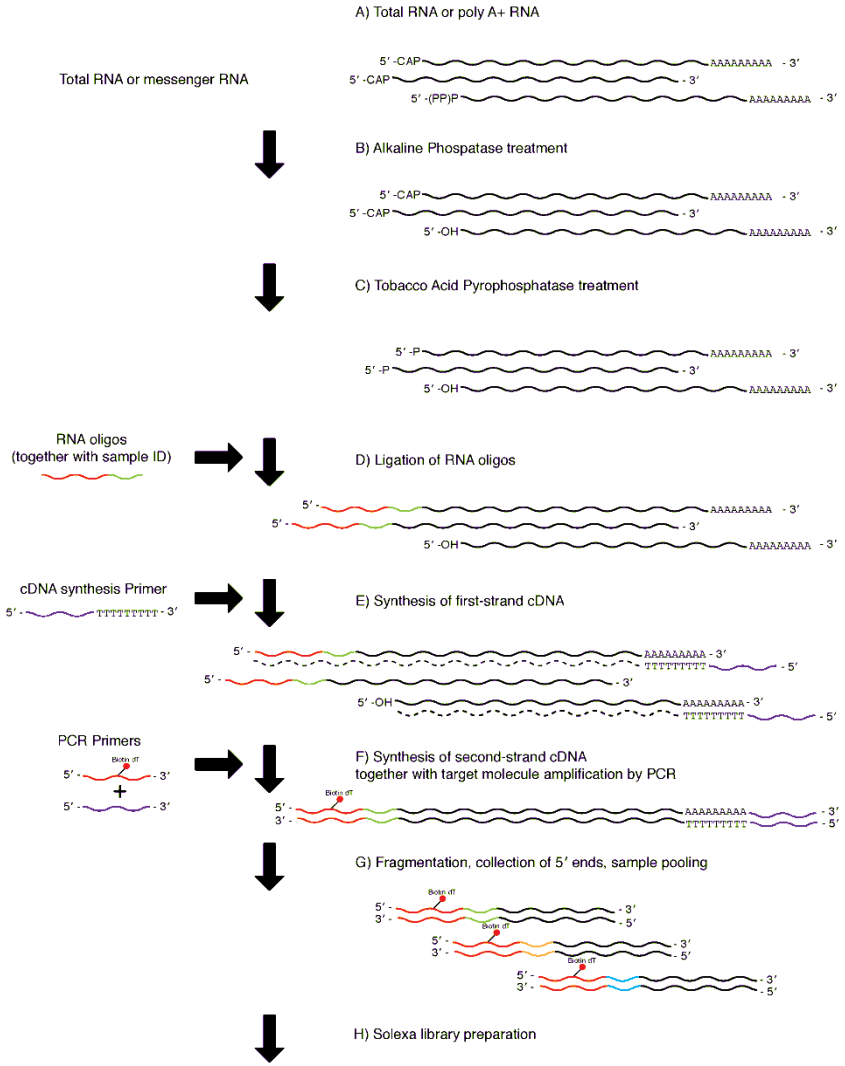
- rRNA is bound by biotinylated oligos
- Double stranded complexes are captured by magnetic beads
- Applicable to bacterial and eukaryotic samples (not relying on polyA tail)

mRNA enrichment



(Machida et al., 2014)

mRNA enrichment



(Machida et al., 2014)

QUESTIONS

- How does illumina sequencing work (overview)?
- Why is bridge amplification important?
- What are possible reasons for sequencing errors?

Quality Control of RNA-Seq data

- RNA-Seq data can be checked by fastQC
- Overrepresented k-mers are caused by multiple reads derived from the same (highly expressed) gene
- GC content of reads is usually higher than in genomic data

EXERCISE & QUESTIONS

- Run fastQC on Col-0 (reference) and 3xmyb FASTQ files!

- Is the technical quality of the RNA-Seq reads ok?
- How long are the RNA-Seq reads?
- Interesting observations?

RNA-Seq read mapping

- Reads (read pairs) represent transcripts
- Reads need to be associated to genes
- Algorithms like BLAST would be way too slow
- Mapping tools like BWA or bowtie are not able to take introns into account (would work for prokaryotes)
- STAR is a dedicated split read mapper: parts of reads can be mapped to different positions on the genome sequence

STAR - reference construction

```
$ STAR \
--runMode genomeGenerate \
--genomeDir /some/directory/ \
--genomeFastaFiles <genome_fasta_file> \
--runThreadN 4 \
--limitGenomeGenerateRAM 40000000000 \
--genomeSAindexNbases 4 \
--sjdbGTFtagExonParentTranscript Parent \
--sjdbGTFfile <reference_gff_file>
```

…. specifies mode to run STAR in

…. output directory

…. ref sequence

… number of CPUs to use

….. use 40GB RAM

… defines size of parts in index

… use annotation

… use annotation

STAR – read mapping

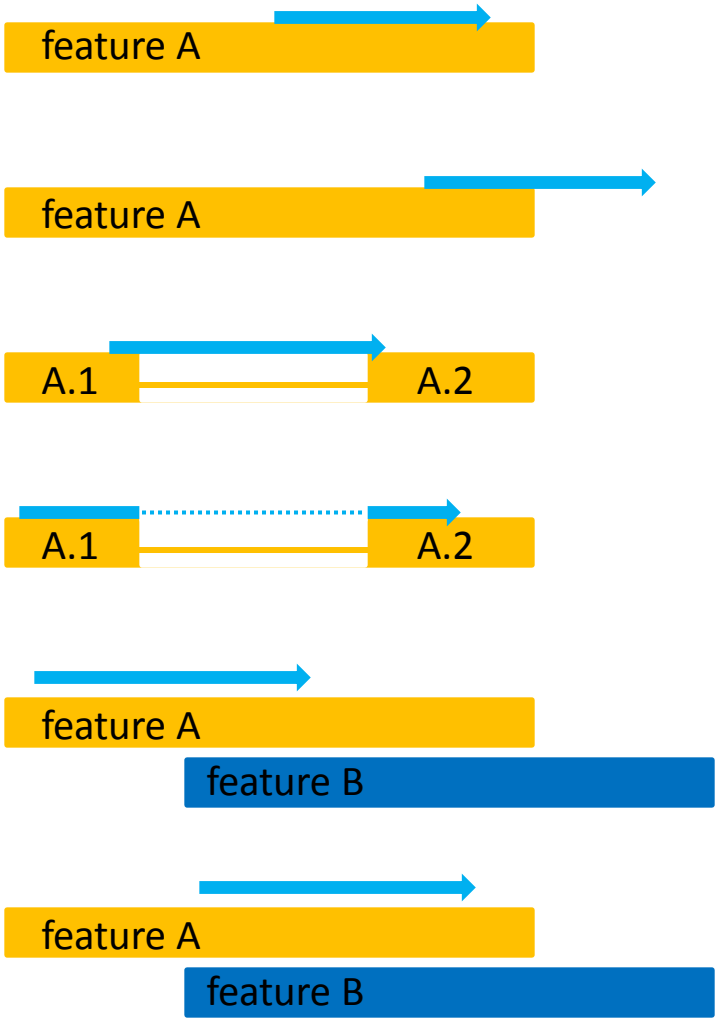
```
$ STAR \
--genomeDir /some/directory \
--readFilesIn <file_fw1.fastq>,<file_fw2.fastq>
   <file_rv1.fastq>,<file_rv2.fastq> \ … PE mapping possible
--readFilesCommand zcat \ … only for compressed input files
--runThreadN 4 \ … number of CPUs to use
--outFileNamePrefix /some/output/directory/ \ … output dir
--limitBAMsortRAM 40000000000 \ … use 40GB RAM
--outBAMsortingThreadN 2 \ … use 2 threads for BAM sorting
--outSAMtype BAM SortedByCoordinate \ … sort BAM
--outFilterMismatchNoverLmax 0.05 \ … max 5% mismatches
--outFilterMatchNminOverLread 0.8  … min 80% read length
```

EXERCISE

- Construct Col-0 (Araport11) reference sequence!
- Run STAR read mapping for both samples separately!

Assigning and counting reads

- Mapped reads need to be assigned to genes
- Reads (or pairs of reads) per gene need to be counted
- Assignment could be done on another feature level as well (transcript, exon, CDS)
- HT-seq and featureCounts are commonly used tools



featureCounts

```
$featureCounts \
-t gene \ … count reads on gene feature level
-g ID \ … ID of feature elements
-a <gff_file> \ … GFF3 file with annotation
-o <output_file> \ … defines output file (.countTable as extension)
<bam_file> … mapping file is input
```

EXERCISE

- Run featureCounts on both mapping files!
- Describe count table format!

Statistical analysis – DESeq2

- Identification of differentially expressed genes requires statistical analysis
- DESeq2 (R package) is one of the most frequently applied tools
- Converting featureCounts output into DESeq2 input files via custom python script
- featureCounts could be used as part of Bioconductor to run all these analysis in R

EXERCISE

- Use the python script ‘construct_DeSeq2_input.py’ to prepare the data for statistical analysis!

WARNING

- To reduce computation costs you are working with only one replicate per genotype while DESeq2 requires some more
- Python script for data preparation will produce three replicates by introducing some pseudo random noise
- **Biological results are artificial!**

DESeq2 input

name genotype

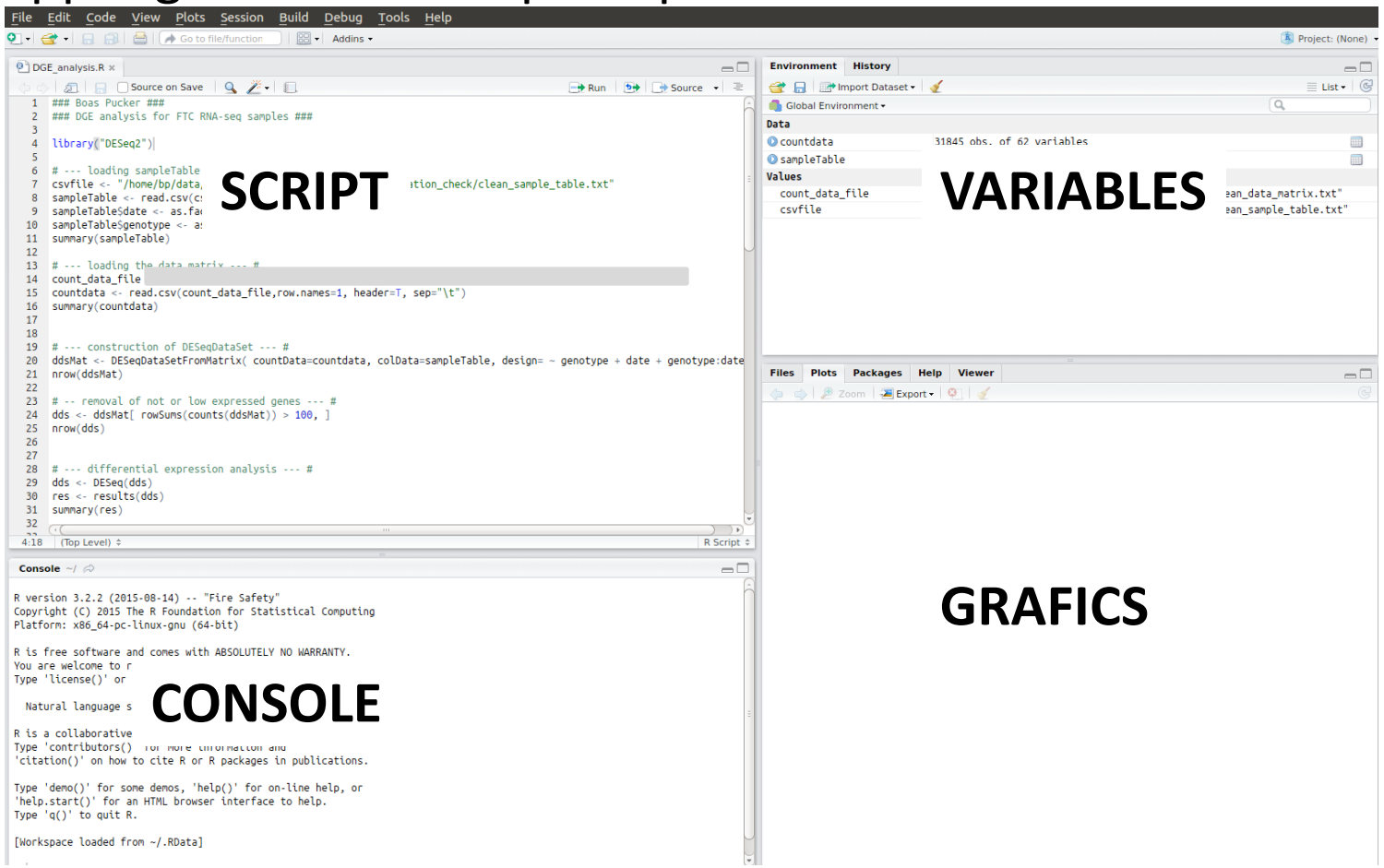
	genotype
Col0_1	Col0
Col0_2	Col0
Col0_3	Col0
myb3x_1	myb3x
myb3x_2	myb3x
myb3x_3	myb3x

More columns are possible e.g. date/time

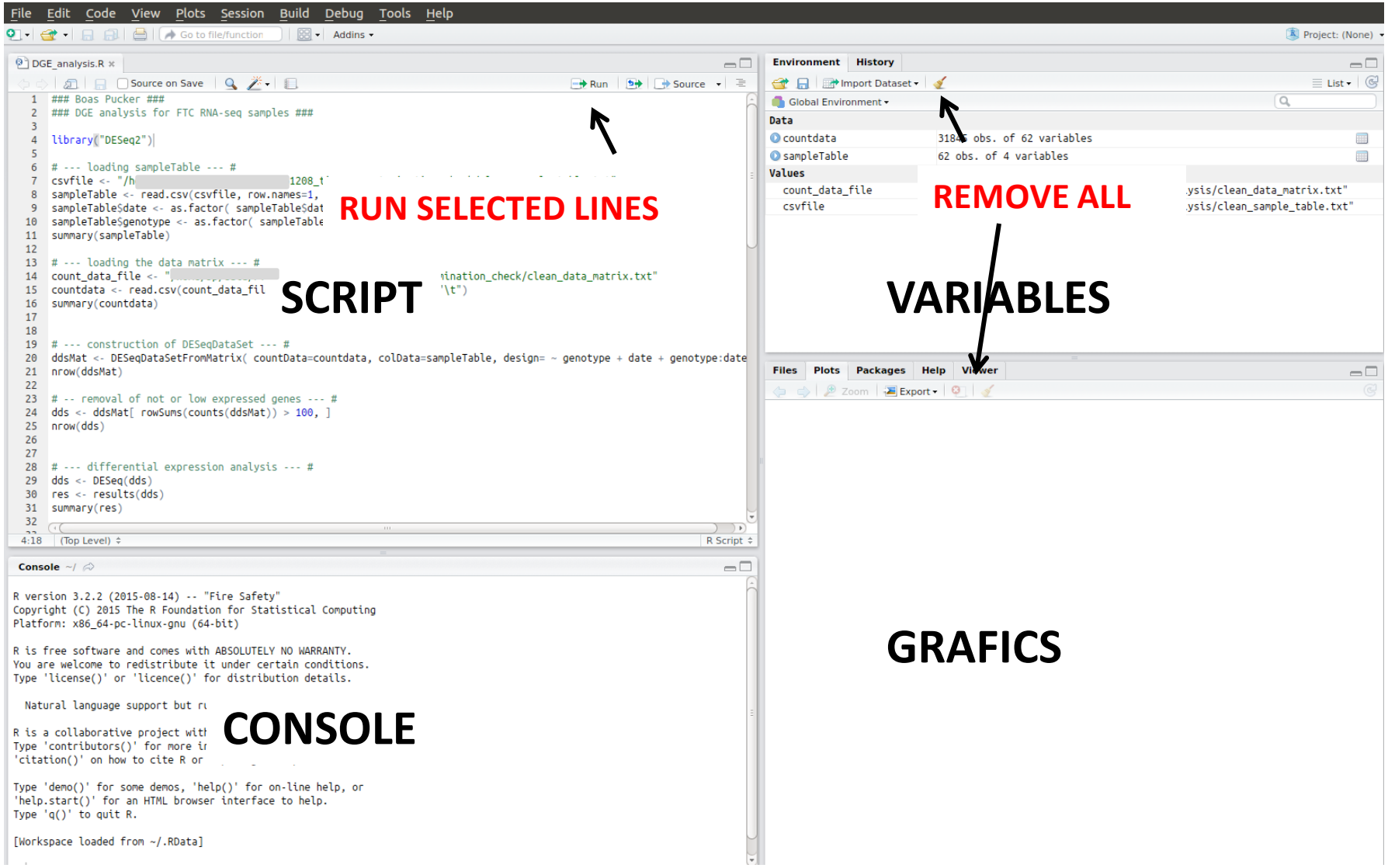
	Col0_1	Col0_2	Col0_3	myb3x_1	myb3x_2	myb3x_3
AT1G01010	215	215	215	102	102	102
AT1G01020	202	202	202	145	145	145
AT1G01030	41	41	41	23	23	23
AT1G01040	318	318	318	180	180	180
AT1G01046	0	0	0	0	0	0
AT1G01050	633	633	633	399	399	399
AT1G01060	427	427	427000	204	204	204
AT1G01070	79	79	79	65	65	65
AT1G01080	453	453	453	366	366	366
AT1G01090	2266	2266	2266	1313	1313	1313
AT1G01100	3258	3258	3258	2351	2351	2351
AT1G01110	73	73	73	48	48	48
AT1G01120	836	836	836	253	253	253
AT1G01130	63	63000	63	25	25	25
AT1G01140	1208	1208	1208	630	630	630
AT1G01150	3	3	3	0	0	0
AT1G01160	133	133	133	134	134	134
AT1G01170	139	139	139	130	130	130
AT1G01180	51	51	51	50	50	50
AT1G01183	0	0	0	0	0	0
AT1G01190	24	24	24	3	3	3
AT1G01200	30	30	30	12	12	12
AT1G01210	153	153	153	82	82	82
AT1G01220	254	254	254	172	172	172
AT1G01225	36	36	36	22	22	22
AT1G01230	259	259	259	188	188	188
AT1G01240	152	152	152	174	174	174
AT1G01250	9	9	9	5	5	5
AT1G01260	264	264	264	120	120	120
AT1G01270	0	0	0	0	0	0
AT1G01280	0	0	0	0	0	0
AT1G01290	161	161	161	95	95	95
AT1G01300	698	698	698	361	361	361

Rstudio

- Tipp: right click on script > open with > Rstudio



Rstudio


File Edit Code View Plots Session Build Debug Tools Help
DGE_analysis.R
Source on Save
Run
Source
Boas Pucker
DGE analysis for FTC RNA-seq samples
library("DESeq2")
# --- loading sampleTable --- #
sampleTable <- read.csv(csvfile, row.names=1,
summary(sampleTable)
# --- loading the data matrix --- #
countdata <- read.csv(count_data_fil
summary(countdata)
# --- construction of DESeqDataSet --- #
ddsMat <- DESeqDataSetFromMatrix(countData=countdata, colData=sampleTable, design= ~ genotype + date + genotype:date
nrow(ddsMat)
# -- removal of not or low expressed genes --- #
dds <- ddsMat[rowSums(counts(ddsMat)) > 100,]
nrow(dds)
# --- differential expression analysis --- #
dds <- DESeq(dds)
res <- results(dds)
summary(res)
RUN SELECTED LINES
SCRIPT
Environment History
Import Dataset
Global Environment
Data
countdata
sampleTable
62 obs. of 4 variables
Values
count_data_file
csvfile
REMOVE ALL
VARIABLES
Files Plots Packages Help Viewer
Zoom
Export
GRAFICS
Console
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but ru
R is a collaborative project with
Type 'contributors()' for more in
'citation()' on how to cite R or
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/.RData]
CONSOLE

DESeq2 script

```
### Boas Pucker ###
### bpucker@cebitec.uni-bielefeld.de ###

### DGE analysis for Arabidopsis RNA-Seq samples in Applied Genome Research course ###

library("DESeq2")

# --- loading sampleTable --- #
csvfile <- "/                                                                    /clean_sample_table.txt"
sampleTable <- read.csv(csvfile, row.names=1, sep="\t")
sampleTable$genotype <- as.factor( sampleTable$genotype )
summary(sampleTable)

# --- loading the data matrix --- #
count_data_file <- "/                                                                    clean_data_matrix.txt"
countdata <- read.csv(count_data_file,row.names=1, header=T, sep="\t")
summary(countdata)

# --- construction of DESeqDataSet --- #
ddsMat <- DESeqDataSetFromMatrix( countData=countdata, colData=sampleTable, design= ~ genotype )
nrow(ddsMat)

# -- removal of not or low expressed genes --- #
dds <- ddsMat[ rowSums(counts(ddsMat)) > 100, ]
nrow(dds)

# --- plot PCA in R studio --- #
rld <- rlog(dds)
ramp <- 1:2/2
cols <- c( rgb(ramp, 0, 0), rgb(0, ramp, 0), rgb(ramp, 0, ramp), rgb(ramp, 0, ramp) )
print ( plotPCA( rld, intgroup=c( "genotype")  ) )

# --- differential expression analysis --- #
dds <- DESeq(dds)
res <- results(dds)
summary(res)

# --- investigate differentially expressed genes --- #
res.05 <- results( dds, alpha=.05  )
table(res.05$padj < .05  )

small.pvalue.index <- head( order( res$padj ), 20  )
names <- row.names(res)
( sig.gene.names <- names[ small.pvalue.index  ] )

outputfile <- "/prj/                                                                    ferentially_expressed_genes.txt"
write( sig.gene.names, outputfile, ncolumns=length( sig.gene.names ), sep="\n"  )

# --- print session information --- #
sessionInfo()

```

DESeq2 script

```
### Boas Pucker ###
### bpucker@cebitec.uni-bielefeld.de ###

### DGE analysis for Arabidopsis RNA-Seq samples in Applied Genome Research course ###

library("DESeq2")

# --- loading sampleTable --- #
csvfile <- "/                                                    /clean_sample_table.txt"
sampleTable <- read.csv(csvfile, row.names=1, sep="\t")
sampleTable$genotype <- as.factor( sampleTable$genotype )
summary(sampleTable)

# --- loading the data matrix --- #
count_data_file <- "/                                                    /clean_data_matrix.txt"
countdata <- read.csv(count_data_file,row.names=1, header=T, sep="\t")
summary(countdata)

```

```
> library("DESeq2")
> csvfile <- "/p                                                    2/clean_sample_table.txt"
> sampleTable <- read.csv(csvfile, row.names=1, sep="\t")
> sampleTable$genotype <- as.factor( sampleTable$genotype )
> summary(sampleTable)
  genotype
 Col0 :3
 myb3x:3
> count_data_file <- "/                                                    /clean_data_matrix.txt"
> countdata <- read.csv(count_data_file,row.names=1, header=T, sep="\t")
> summary(countdata)
     Col0_1            Col0_2            Col0_3            myb3x_1           myb3x_2           myb3x_3
 Min.   :      0   Min.   :       0   Min.   :       0   Min.   :      0   Min.   :      0   Min.   :      0
 1st Qu.:      0   1st Qu.:       0   1st Qu.:       0   1st Qu.:      0   1st Qu.:      0   1st Qu.:      0
 Median :     28   Median :      28   Median :      28   Median :     18   Median :     18   Median :     18
 Mean   :   2208   Mean   :    2441   Mean   :    3056   Mean   :   1825   Mean   :   1352   Mean   :   1573
 3rd Qu.:    205   3rd Qu.:     205   3rd Qu.:     205   3rd Qu.:    139   3rd Qu.:    139   3rd Qu.:    139
 Max.   :3444000   Max.   :12461000   Max.   :13316000   Max.   :7151000   Max.   :2088000   Max.   :8294000
```

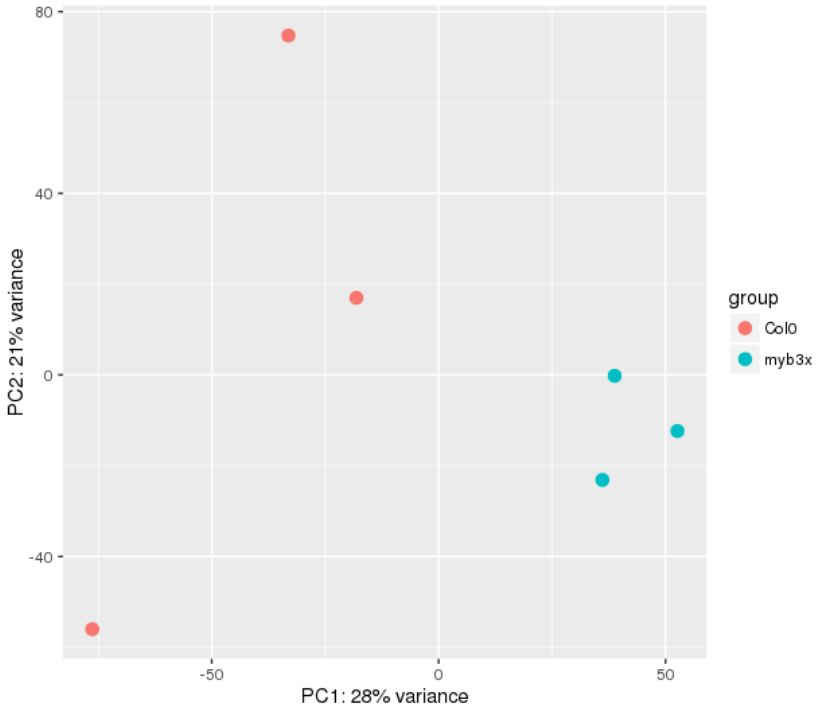
DESeq2 script

```
### Boas Pucker ###
### bpucker@cebitec.uni-bielefeld.de ###

### DGE analysis for Arabidopsis RNA-Seq samples in Applied Genome Research course ###

library("DESeq2")

# --- loading sampleTable --- #
csvfile <- "/                                                    /clean_sample_table.txt"
sampleTable <- read.csv(csvfile, row.names=1, sep="\t")
sampleTable$genotype <- as.factor( sampleTable$genotype )
summary(sampleTable)

# --- loading the data matrix --- #
count_data_file <- "/                                                    clean_data_matrix.txt"
countdata <- read.csv(count_data_file,row.names=1, header=T, sep="\t")
summary(countdata)

# --- construction of DESeqDataSet --- #
ddsMat <- DESeqDataSetFromMatrix( countData=countdata, colData=sampleTable, design= ~ genotype )
nrow(ddsMat)

# -- removal of not or low expressed genes --- #
dds <- ddsMat[ rowSums(counts(ddsMat)) > 100, ]
nrow(dds)
```

```
> ddsMat <- DESeqDataSetFromMatrix( countData=countdata, colData=sampleTable, design= ~ genotype )
> nrow(ddsMat)
[1] 33296
> # -- removal of not or low expressed genes --- #
> dds <- ddsMat[ rowSums(counts(ddsMat)) > 100, ]
> nrow(dds)
[1] 17672
>
```

DESeq2 script

```
### Boas Pucker ###
### bpucker@cebitec.uni-bielefeld.de ###

### DGE analysis for Arabidopsis RNA-Seq samples in Applied Genome Research course ###

library("DESeq2")

# --- loading sampleTable --- #
csvfile <- "/                                                    /clean_sample_table.txt"
sampleTable <- read.csv(csvfile, row.names=1, sep="\t")
sampleTable$genotype <- as.factor( sampleTable$genotype )
summary(sampleTable)

# --- loading the data matrix --- #
count_data_file <- "/
countdata <- read.csv(count_data_file,row.names=1, header=T, sep="\t")
summary(countdata)

# --- construction of DESeqDataSet --- #
ddsMat <- DESeqDataSetFromMatrix( countData=countdata, colData=sampleTable, design
nrow(ddsMat)

# -- removal of not or low expressed genes --- #
dds <- ddsMat[ rowSums(counts(ddsMat)) > 100, ]
nrow(dds)

# --- plot PCA in R studio --- #
rld <- rlog(dds)
ramp <- 1:2/2
cols <- c( rgb(ramp, 0, 0), rgb(0, ramp, 0), rgb(ramp, 0, ramp), rgb(ramp, 0, ramp
print ( plotPCA( rld, intgroup=c( "genotype")  ) )

# --- differential expression analysis --- #
dds <- DESeq(dds)
res <- results(dds)
summary(res)

# --- investigate differentially expressed genes --- #
res.05 <- results( dds, alpha=.05  )
table(res.05$padj < .05  )

small.pvalue.index <- head( order( res$padj ), 20  )
names <- row.names(res)
( sig.gene.names <- names[ small.pvalue.index  ] )

outputfile <- "/                                                    /di
write( sig.gene.names, outputfile, ncolumns=length( sig.gene.names ), sep="\n"  )

# --- print session information --- #
sessionInfo()
```



DESeq2 script

```
> # --- differential expression analysis --- #
> dds <- DESeq(dds)
estimating size factors
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
-- note: fitType='parametric', but the dispersion trend was not well captured by the
   function: y = a/x + b, and a local regression fit was automatically substituted.
   specify fitType='local' or 'mean' to avoid this message next time.
final dispersion estimates
fitting model and testing
> res <- results(dds)
> summary(res)

out of 17672 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)     : 6, 0.034%
LFC < 0 (down)   : 6, 0.034%
outliers [1]     : 1268, 7.2%
low counts [2]   : 0, 0%
(mean count < 15)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

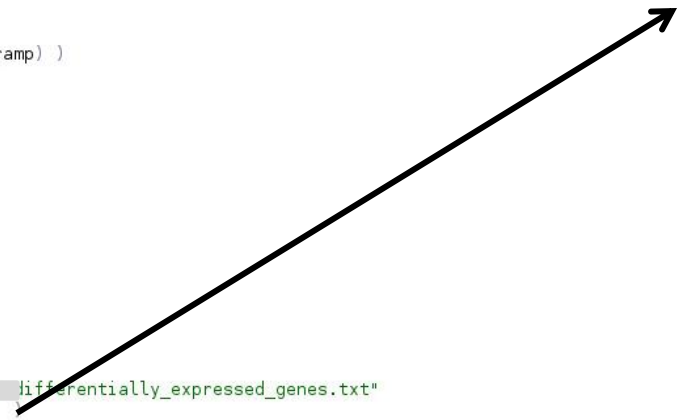
DESeq2 script

```
32
33 # --- differential expression analysis --- #
34 dds <- DESeq(dds)
35 res <- results(dds)
36 summary(res)
37
38 # --- investigate differentially expressed genes --- #
39 res.05 <- results( dds, alpha=.05  )
40 table(res.05$padj < .05  )
41
42 small.pvalue.index <- head( order( res$padj ), 20  )
43 names <- row.names(res)
44 ( sig.gene.names <- names[ small.pvalue.index  ] )
45
46 outputfile <- "/                                                    /differentially_expressed_genes.txt"
47 write( sig.gene.names, outputfile, ncolumns=length( sig.gene.names ), sep="\n"  )
48
49 # --- print session information --- #
50 sessionInfo()
51
```

```
> res.05 <- results( dds, alpha=.05  )
> table(res.05$padj < .05  )

FALSE  TRUE
16392    12
> small.pvalue.index <- head( order( res$padj ), 20  )
> names <- row.names(res)
> ( sig.gene.names <- names[ small.pvalue.index  ] )
 [1] "AT1G09415" "AT3G15730" "AT3G22440" "AT5G46100" "AT5G01630" "AT1G28680" "AT2G19750" "AT3G60210" "AT5G01980" "AT5G26330"
[11] "AT5G61450" "AT2G01010" "AT1G01010" "AT1G01020" "AT1G01030" "AT1G01040" "AT1G01050" "AT1G01070" "AT1G01080" "AT1G01090"
>
```

DESeq2 script

```
### Boas Pucker ###
### bpucker@cebitec.uni-bielefeld.de ###

### DGE analysis for Arabidopsis RNA-Seq samples in Applied Genome Research course ###

library("DESeq2")

# --- loading sampleTable --- #
csvfile <- "/p                                                                    clean_sample_table.txt"
sampleTable <- read.csv(csvfile, row.names=1, sep="\t")
sampleTable$genotype <- as.factor( sampleTable$genotype )
summary(sampleTable)

# --- loading the data matrix --- #
count_data_file <- "/p                                                                    clean_data_matrix.txt"
countdata <- read.csv(count_data_file,row.names=1, header=T, sep="\t")
summary(countdata)

# --- construction of DESeqDataSet --- #
ddsMat <- DESeqDataSetFromMatrix( countData=countdata, colData=sampleTable, design= ~ genotype )
nrow(ddsMat)

# -- removal of not or low expressed genes --- #
dds <- ddsMat[ rowSums(counts(ddsMat)) > 100, ]
nrow(dds)

# --- plot PCA in R studio --- #
rld <- rlog(dds)
ramp <- 1:2/2
cols <- c( rgb(ramp, 0, 0), rgb(0, ramp, 0), rgb(ramp, 0, ramp), rgb(ramp, 0, ramp) )
print ( plotPCA( rld, intgroup=c( "genotype")  ) )

# --- differential expression analysis --- #
dds <- DESeq(dds)
res <- results(dds)
summary(res)

# --- investigate differentially expressed genes --- #
res.05 <- results( dds, alpha=.05  )
table(res.05$padj < .05  )

small.pvalue.index <- head( order( res$padj ), 20  )
names <- row.names(res)
( sig.gene.names <- names[ small.pvalue.index  ] )

outputfile <- "/p                                                                    differentially_expressed_genes.txt"
write( sig.gene.names, outputfile, ncolumns=length( sig.gene.names ), sep="\n"  )

# --- print session information --- #
sessionInfo()

```



```
AT1G09415
AT3G15730
AT3G22440
AT5G46100
AT5G01630
AT1G28680
AT2G19750
AT3G60210
AT5G01980
AT5G26330
AT5G61450
AT2G01010
AT1G01010
AT1G01020
AT1G01030
AT1G01040
AT1G01050
AT1G01070
AT1G01080
AT1G01090
```

DESeq2 script

```
49 # --- print session information --- #
50 sessionInfo()
51
```

```
> # --- print session information --- #
> sessionInfo()
R version 3.3.1 (2016-06-21)
Platform: x86_64-redhat-linux-gnu (64-bit)
Running under: Fedora 23 (Twenty Three)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8    LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel  stats4    stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] DESeq2_1.12.3               SummarizedExperiment_1.2.3 Biobase_2.32.0             GenomicRanges_1.24.1
[5] GenomeInfoDb_1.8.1          IRanges_2.6.0              S4Vectors_0.10.1           BiocGenerics_0.18.0

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.5          RColorBrewer_1.1-2   plyr_1.8.4           XVector_0.12.0       tools_3.3.1          zlibbioc_1.18.0
 [7] digest_0.6.9         rpart_4.1-10         RSQLite_1.0.0        annotate_1.50.0      gtable_0.2.0         lattice_0.20-33
[13] Matrix_1.2-6         DBI_0.4-1            gridExtra_2.2.1      genefilter_1.54.2    cluster_2.0.4        locfit_1.5-9.1
[19] grid_3.3.1           nnet_7.3-12          data.table_1.9.6     AnnotationDbi_1.34.3 XML_3.98-1.4         survival_2.39-4
[25] BiocParallel_1.6.2   foreign_0.8-66       latticeExtra_0.6-28  Formula_1.2-1        geneplotter_1.50.0   ggplot2_2.1.0
[31] Hmisc_3.17-4         scales_0.4.0         splines_3.3.1        colorspace_1.2-6     xtable_1.8-2         labeling_0.3
[37] acepack_1.4.0        munsell_0.4.3        chron_2.3-47
> |
```

EXERCISE

- Run DESeq2 script via Rstudio to identify differentially expressed genes!

Annotation of resulting AGIs

- Functional annotation can be mapped from TAIR10/Araport11
- Python script ‘map_annotation.py’ can be applied for this purpose
- Results can be interpreted based on description of gene functions

Adjusted p-value

- RNA-Seq analysis usually compare several thousand genes => multiple testing
- Each test is associated with 5% error rate
- Example:
  - 30,000 genes * 0.05 = 600 false positives
- p-value correction (padj):
  - multiply p-value by number of tests (=number of genes)

EXERCISE & QUESTIONS

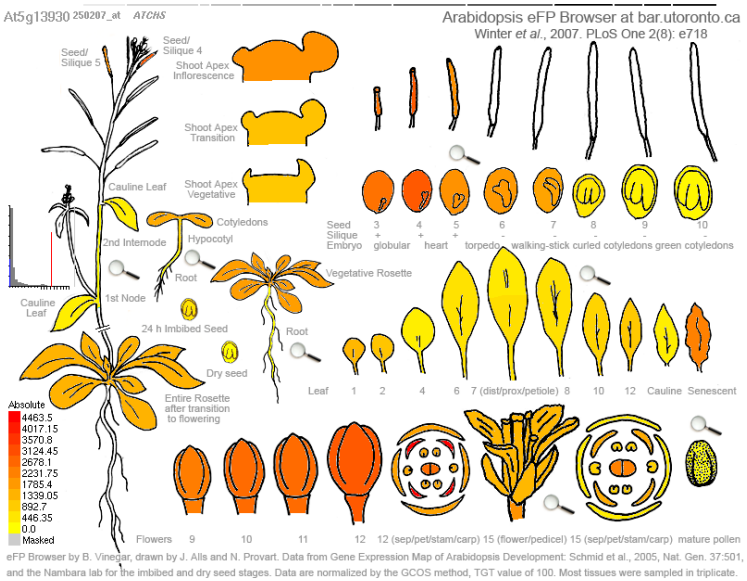
- Run ‘map_annotation.py’ on differentially expressed gene set!

- Which genes are differentially expressed?
- What is there function?

Validation of RNA-Seq results

- Check available resources:

http://bar.utoronto.ca/efp/cgi-bin/efpWeb.cgi



- Use orthogonal methods (e.g. RT-qPCR)

EXERCISE

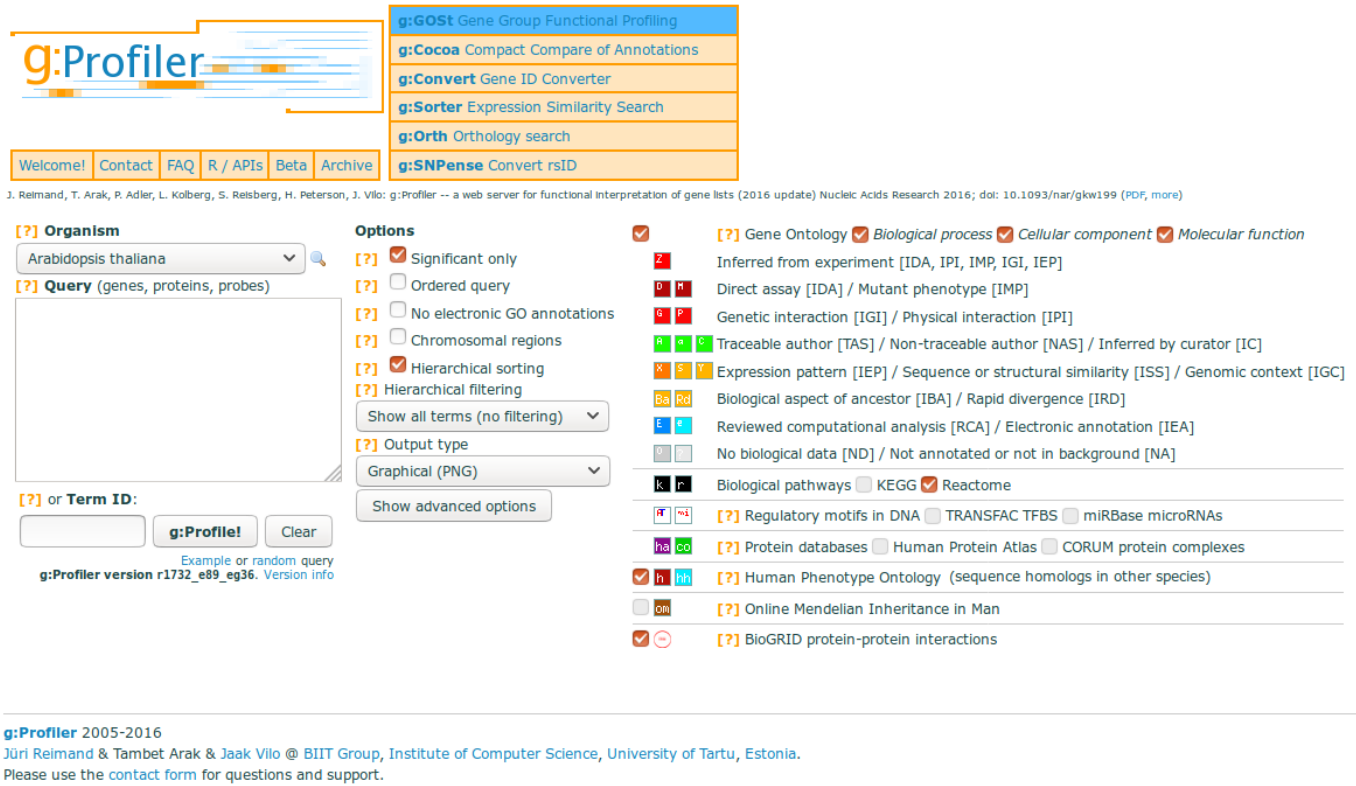
- Check the expression patterns of differentially expressed genes!
- Are your genes expressed in leaves?

GO terms

- GO = gene ontology
- System for assignment of functional annotations
- Defined vocabulary
- Enrichment of GOs in a gene set indicates up- or down-regulation
- Tools for GO enrichment:
  - BINGO, Gorilla, gProfiler, Ontologizer, VLAD

GO term enrichment analysis

- g:profiler: http://biit.cs.ut.ee/gprofiler/



EXERCISE

- Check the differentially expressed genes for enriched GO terms by running g:profiler on the AGIs!
- Get AGIs of flavonoid biosynthesis genes (CHS, CHI, DFR, ANS, FLS) and use them as control!